JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 25 No. 5, pp. 27-36, May 2020
https://doi.org/10.9708/jksci.2020.25.05.027

Research of Semantic Considered Tree Mining Method for an Intelligent Knowledge-Services Platform

Juryon Paik*

*Professor, Dept. of Digital Information and Statistics, Pyeongtaek University, Pyeongtaek, Korea

[Abstract]

In this paper, we propose a method to derive valuable but hidden infromation from the data which is the core foundation in the 4th Industrial Revolution to pursue knowledge-based service fusion. The hyper-connected societies characterized by IoT inevitably produce big data, and with the data in order to derive optimal services for trouble situations it is first processed by discovering valuable information. A data-centric IoT platform is a platform to collect, store, manage, and integrate the data from variable devices, which is actually a type of middleware platforms. Its purpose is to provide suitable solutions for challenged problems after processing and analyzing the data, that depends on efficient and accurate algorithms performing the work of data analysis. To this end, we propose specially designed structures to store IoT data without losing the semantics and provide algorithms to discover the useful information with several definitions and proofs to show the soundness.

▸ Key words: Knowledge-based platform, Unstructured data, Tree data, Bits representation, Binary code, Pairsets

[요 약]

본 논문은 지식기반의 서비스 융합을 추구하는 4차산업혁명의 핵심 기반인 *데이터*로부터 유용하지만 드러나지 않는 정보들을 추출하는 방식을 제안한다. IoT로 대표되는 초연결사회에서 빅데이터의 생성은 필연적이며 그로부터 최적의 서비스를 도출하기 위해서는 가치있는 데이터를 찾아내는 것은 최우선으로 수행되어야 한다. 다양한 디바이스로부터 엄청난 양의 데이터를 수집·저장·관리하고 통합하는 데이터중심 IoT 플랫폼은 일종의 미들웨어 솔루션으로, 플랫폼의 궁극적인 목적은 빅데이터를 적시적소에 맞게 가공 및 분석수행 후 가치 있는 결과를 도출하여 최적의 답안을 제시하는 것이다. 이는 데이터를 분석하는 효율적이고 정확한 알고리즘을 필요로 한다. 이를 위해 본 논문은 분산되어 생성되는 IoT 데이터로부터 유용 정보 추출을 위해 시맨틱을 고려하여 원데이터를 저장하는 특화된 구조체를 설계하고 제안한 구조체에 기반하여 가치있는 정보를 찾아내기 위한 알고리즘을 다양한 정의와 증명을 사용하여 제시한다.

▸ **주제어**: 지식기반, 비정형데이터, 트리데이터, 비트표현, 이진코드, 페어세트

• First Author: Juryon Paik, Corresponding Author: Juryon Paik
*Juryon Paik (jrpaik@ptu.ac.kr), Dept. of Digital Information and Statistics, Pyeongtaek University
• Received: 2020. 04. 22, Revised: 2020. 05. 18, Accepted: 2020. 05. 20.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

I. Introduction

초연결·초융합·초지능이라는 3가지 키워드로 요약할 수 있는 4차산업혁명은 결국 지식기반의 서비스 융합을 추구하는 산업으로의 확장과 그 확장의 핵심이자 기반은 '데이터'이며 그 데이터로부터 최적의 의사결정을 도출하는 기술이라고 할 수 있다. 빅데이터는 IoT(Internet of Things, 사물인터넷)로 대표되는 초연결사회에서 필연적으로 생성될 수밖에 없다. IoT의 수많은 센서 기반의 네트워크는 빠르고 지속적으로 수많은 정보를 교환하며 과거와는 비교할 수 없을 정도로 대량의 비정형 데이터를 생산하기 때문이다. IoT, 빅데이터, 인공지능, 클라우드 등 4차산업혁명의 여러 기술들과 결합되면서 스마트홈, 스마트빌딩, 스마트팩토리, 스마트시티 등 다양한 분야에서 영향력을 강화해 나가고 있으며 새로운 시장을 창출해나가려 하고 있다.

문제는 IoT가 수많은 디바이스들로 구성될 뿐만 아니라, 이를 연결하고 데이터를 수집하기 위한 네트워크, 수집한 데이터를 저장하고 분석하기 위한 클라우드와 인공지능 기술, 그리고 분석된 결과를 활용하기 위한 응용프로그램 등 복잡한 구조를 갖추고 있기 때문에 이를 구축하고 개발하는 것이 쉽지 않은 상황이라는 것이다. 이를 해결하기 위한 하나의 솔루션으로 주목받고 있는 기술이 IoT 플랫폼이다.

IoT 플랫폼은[1,2] 물리적인 객체를 온라인에 구현하기 위해 필요한 요소를 제공하는, 즉 IoT 환경 구축과 관리를 효율적으로 하기 위한 일종의 미들웨어로써 기업들은 이를 통해 표준화된 구성 요소를 조합함으로써 IoT 환경을 구현할 수 있기 때문에 IoT 솔루션 개발 비용과 시간을 크게 단축할 수 있다. 그 중 특히 다양한 디바이스로부터 획득하는 엄청난 양의 데이터를 수집·저장·분석하고 통합 관리하는 **데이터 중심(Data-Centric) IoT 플랫폼**은 시맨틱 기술을 접목한 사물 연동을 통한 데이터 상호운용과 지능형 서비스 제공을 목적으로 제3의 비즈니스 창출을 추구한다[3].

데이터중심 IoT 플랫폼의 궁극적인 목적은 생성된 빅데이터를 적시적소에 맞게 가공 및 분석을 수행하여 가치 있는 결과를 도출하여 최적의 답안을 제시하는 것으로 사물에서 수집하는 데이터의 가치는 데이터를 분석하는 효율적이고 정확한 알고리즘에 기인한다고 할 수 있다. 본 논문은 분산되어 생성되는 IoT 데이터로부터 시맨틱을 고려한 유용 정보 추출을 수행하는 방식 및 알고리즘을 제시하고 해당 방법을 모듈로 적용한 지능형 지식서비스를 위한 플랫폼 설계를 제안한다.

II. Preliminaries

1. JSON(JavaScript Object Notation)

IoT를 통해 수집되는 데이터들을 표현하는 가장 대중적인 방식들 중의 하나는 JSON을 사용하여 저장하는 것이다. 데이터 범람의 시작이라 할 수 있는 웹 데이터의 대표적인 표현·전송·저장 방식은 XML이다. XML의 역할을 IoT에서는 JSON이 담당한다고도 할 수 있다. 데이터 표현에 있어서의 유연성, 손쉬운 데이터 교환 그리고 경량은 JSON의 사용을 촉진했으며 센서를 통해 데이터를 전송 및 교환하는 IoT 환경에서 매우 일반적인 인코딩 방식으로 사용되게 된 것이다. Table 1은 JSON을 사용하여 특정 사람에 대한 정보 일부를 인코딩했을 때 해당 정보의 계층 관계를 표로 나타낸 것이다.

Table 1. JSON Encoded Person Data

firstName	"John"	
lastName	"Smith"	
age	25	
address	streetAddress	"21 2nd Street"
	city	"New York"
	state	"NY"
	postalCode	"10021-3100"
gender	type	"male"
spouse	null	
phoneNumbers	[	
	type	"home"
	number	"2125551234"
	type	"office"
	number	"646 5554567"
	type	"mobile"
	number	"123 456 7890"
	]	

XML과 JSON 모두 쉬운 사용 그리고 어떤 종류의 데이터도 표현할 수 있는 유연성을 대표적 특징으로 하는데 이는 두 방식 모두 트리라고 하는 동일한 구조를 가지고 있기 때문이다[4]. 본 논문은 트리 구조를 갖는 대량의 JSON 타입 데이터들을 대상으로 시맨틱 고려한 표현 방식을 제안하고 그에 기반을 두어 트리 마이닝을 수행하는 알고리즘을 일차적으로 제안한다.

2. Structure of trees

비구조적·비구조화 데이터인 비정형 데이터들에 정형성을 부여하기 위해 사용되는 방식 중 하나인 트리 구조화는 비정형 데이터들의 다양성과 정보 손실을 최소화하면서 표현의 유연성을 최대화로 보장하는 방식이다. 이질성과 다양성의 보장은 트리 구조가 갖는 계층적 구조로 인한 데

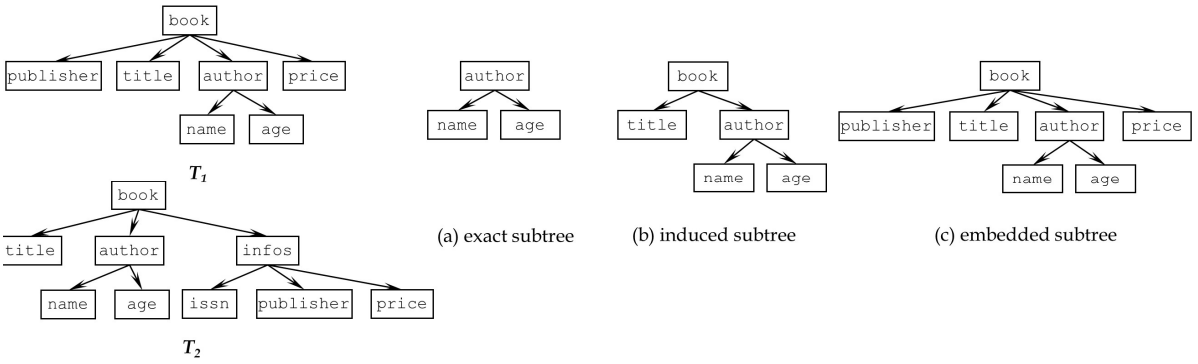

Fig. 1. Different Types of Subtrees

이터 분할성(granularity) 향상에 기인하며 대다수의 비정형 데이터들은 트리로 충분히 표현 가능하다. 논문 [5,6,7]에 근거하여 트리의 대표적인 정의들을 살펴본다.

(정의 1) 루트 트리

노드와 노드는 방향을 갖는 간선에 의해 연결되며 사이클이 존재하지 않는 트리로 다음의 특징을 갖는다. (1) 트리의 시작은 언제나 루트(root)라 불리는 특별한 노드로부터 시작되며 루트 노드로 들어오는 간선은 존재하지 않으며 나가는 간선만 존재한다. (2) 루트 노드를 제외한 다른 노드들의 들어오는 간선의 수는 하나이다. (3) 루트 노드부터 각 노드 사이에 존재하는 경로는 유일하다.

(정의 2) 레이블 트리

단일 트리 $T = (r, N, E, L)$ 가 주어졌을 때, 트리 T의 루트 노드는 r이며, $N = \{v_1, v_2, ..., v_i\}$ 은 T를 구성하고 있는 노드 전체의 집합이며, $E = \{(u, v) \mid u, v \in N\}$ 는 노드와 노드를 연결하는 간선들의 집합이다. 이 때, 집합 N에 속하는 각 노드들은 집합 L에 속하는 단일 레이블을 갖는다.

III. Tree Mining Definitions

1. Subtrees

트리 구조의 대량 데이터로부터 유용 정보를 추출한다는 것은 빈번하게 자주 발생하는 서브트리를 유도한다는 것으로 서브트리의 개념은 제한 범위에 따라 몇 가지로 나뉜다.

(정의 3) 서브트리

두 개의 트리 $T = (r, N, E, L)$와 $S = \{r_S, N_S, E_S, L_S\}$가 주어졌을 때, 트리 T가 S의 인스턴스를 포함하고 있으면 트리 S는 트리 T의 서브트리라고 칭한다. 이 때, S를 구성하고 있는 노드들의 관계가 트리 T 내에서 부모-자식 관계인지 조상-후손 관계인지에 따라 exact 서브트리, induced 서브트리, embedded 서브트리로 나뉜다.

(1) 트리 S의 인스턴스가 주어진 다음 조건들을 모두 충족한다면 트리 S는 트리 T의 *exact 서브트리*이다. ① $N_S \subseteq N$, ② $E_S \subseteq E$, ③ 임의의 노드 $v \in N$가 $v \in N_S$ 라면 노드 v의 모든 후손 노드들 또한 N_S의 원소이다. ④ 집합 E_S의 원소인 모든 간선 (u, v)에 대해서 노드 u와 노드 v가 부모-자식 관계를 갖는다면 트리 T에서도 간선 (u, v)의 관계는 동일한 부모-자식 관계를 갖는다. ⑤ N_S에 속하는 임의의 노드 v의 레이블은 집합 L과 L_S에 모두 속한다.

(2) 트리 S의 인스턴스가 주어진 다음 조건들을 모두 충족한다면 트리 S는 트리 T의 *induced 서브트리*이다. ① $N_S \subseteq N$, ② $E_S \subseteq E$, ③ 집합 E_S의 원소인 모든 간선 (u, v)에 대해서 노드 u와 노드 v가 부모-자식 관계를 갖는다면 트리 T에서도 간선 (u, v)의 관계는 동일한 부모-자식 관계를 갖는다. ④ N_S에 속하는 임의의 노드 v의 레이블은 집합 L과 L_S에 모두 속한다.

(3) 트리 S의 인스턴스가 주어진 다음 조건들을 모두 충족한다면 트리 S는 트리 T의 *embedded 서브트리*이다. ① $N_S \subseteq N$, ② 집합 E_S의 원소인 모든 간선 (u, v)는 트리 T 내에서 노드 u는 노드 v의 조상 노드로 위치하며, ③ N_S에 속하는 임의의 노드 v의 레이블은 집합 L과 L_S에 모두 속한다.

서브트리 탐색이 상대적으로 수월하지만 너무 많은 제약으로 연구의 발전성이 낮은 서브트리는 exact 서브트리이며 embedded 서브트리는 exact 서브트리와 induced 서브트리를 확장한 개념으로 인스턴스 S의 노드와 노드 사이의 관계를 조상-후손으로 확대하여 서브트리의 제약사항을 완화한 개념이다. 본 논문에서 대상으로 하는 서브트리 또한 embedded 서브트리로 단어의 편의성을 위해 서브트리로 지칭한다. Fig. 1은 도서 관련 데이터를 저장하고 있는 두 개의 서로 다른 트리 T_1과 T_2로부터 (정의 3)에서 제시한 세 가지 서로 다른 구조를 갖는 서브트리를 추출한 결과를 보인다.

(정의 4) 빈발도

임의의 트리 S가 대상 트리 T에 인스턴스로 존재할 경우 S의 빈발도 $freq_T(S)$는 두 가지 경우로 나누어 계산될 수 있다. (1) 인스턴스 S가 트리 T에 존재하는지 또는 존재하지 않는지에 중점을 둘 경우 $freq_T(S)$의 값은 1 아니면 0으로 계산된다. 동일 트리 T에서 인스턴스 S가 한 개이던 여러 개이던 $freq_T(S)$의 값은 1로 동일하다. (2) 인스턴스 S의 발생빈도에 중점을 두는 경우, 동일 트리에서 여러 번 발생할 경우의 $freq_T(S)$ 값이 더 높다. 특정 트리에서 서브트리 S가 얼마나 자주 발생하는지에 중점을 둔다. 본 논문에서는 전자의 방식을 적용한 빈발도를 사용하여 의미기반 트리 마이닝 방법을 제안 후 차후 연구에서 후자의 방법을 적용하여 확장하고자 한다.

2. Extraction of subtrees

트리 마이닝을 위해서 일반적으로 적용되는 방법들 중 하나는 대량의 트리 데이터들 내에서 인스턴스로 빈발하게 내재되어 있는 서브트리를 발견하는 것이다. n 개의 트리 데이터가 저장된 데이터베이스 D 가 있다고 가정하면, $D = \{T_1, T_2, \ldots T_n\}$으로 표기되며 전체 데이터베이스 D의 크기 $|D| = \sum_{i=1}^{n} T_i$ 가 된다.

(정의 5) 지지도

트리 데이터베이스 D 에 대해서 임의의 트리 S의 빈발도 $freq_D(S)$는 정의 4와 데이터베이스 D의 크기에 의해서 $\sum_{i=1}^{n} freq_{T_i}(S)$로 정의된다. 만약 대상 트리 T_i에 인스턴스 S가 존재한다면 $freq_{Ti}(S)$는 1, 존재하지 않는다면 0이 된다. 데이터베이스 D에 대한 트리 S의 지지도(support), $sup_D(S)$, 는 D에 저장된 전체 트리 데이터 중에서 트리 S를 서브트리로 포함하고 있는 트리들의 비를 의미하며

$$sup_D(S) = \frac{freq_D(S)}{|D|} = \frac{\sum_{i=1}^{n} freq_{T_i}(S)}{\sum_{i=1}^{n} T_i}$$ 공식으로 계산된다.

임의의 트리 S의 지지도 값이 주어진 지지도의 기준 값보다 크거나 같다면 해당 트리 S는 데이터베이스 D 내에서 빈발하게 발생하는 서브트리로 정의하며 빈발 서브트리라고 지칭한다. 이 때, 마이닝 작업을 수행하는 사용자에 의해 주어지는 최소한의 지지도 값을 최소지지도(minimum support)라 하며 $minsup$ 또는 σ로 표기한다. 주어진 최소지지도 σ 값 이상의 지지도 값을 갖는 서브트리들을 σ-빈발서브트리라 한다.

(정의 6) 최대 빈발 서브트리

σ 값이 주어지고 임의의 트리 S가 다음 두 조건을 모두 충족한다면 S는 D에 대한 σ-최대빈발서브트리라 한다. (1) $sup_D(S) \geq \sigma$, (2) 데이터베이스 D 내에는 S를 σ-빈발서브트리로 포함하는 또 다른 σ-빈발서브트리 S'은 존재하지 않는다.

n개의 노드들로 구성된 트리의 경우 최대 2^n 개의 서브트리 생성이 가능하다. 데이터베이스 D를 구성하고 있는 트리 데이터 사이즈가 작다면 σ 값 이상의 지지도를 갖는 서브트리들을 추출하는 것은 허용 가능한 작업이지만 만일 트리 데이터들의 크기가 지속적으로 그리고 빠르게 증가하는 스토리지의 경우 최악의 경우 전체 트리들의 노드 총합의 지수승만큼 빈발서브트리 추출이 이루어져야 하는데 이는 사실상 계산이 불가능할 수도 있다. 또한 중복 빈발서브트리 추출의 문제점도 발생한다.

최대빈발서브트리는 자신이 빈발서브트리면서 자신의 모든 서브트리들이 빈발서브트리가 되는 서브트리로 해당 데이터베이스의 σ-최대빈발서브트리들로부터 모든 σ-빈발서브트리들을 추출할 수 있다. σ-최대빈발서브트리들의 개수는 σ-빈발서브트리들의 개수보다 적기 때문에 대량의 트리 데이터 마이닝에 더 적합하다고 할 수 있다.

Fig. 2는 전자제품에 대한 서로 다른 정보를 담고 있는 세 개의 트리 데이터로부터 minsup에 따른 최대빈발서브트리와 빈발서브트리가 어떻게 도출되는지를 보이는 예로 추출된 서브트리 개수에 있어서 그 차이가 확연한 것을 알 수 있다. 또한 2/3-최대빈발서브트리의 경우 7개의 빈발서브트리 모두 포함되어 있는 것을 알 수 있다.

IV. Semantic-based Tree Mining Approach

트리로 구조화된 다양한 비정형 데이터들로부터 유용하지만 감춰져 있는 정보들, 즉 빈발서브트리 또는 최대빈발서브트리들을 직접적으로 추출할 수 없다. 일반적으로 트리 구조 자체를 연산이 수월한 인접리스트[8, 10], 인접매트릭스[9, 10, 11], 또는 스트링[12, 13] 등을 사용하여 원 정보의 손실 없이 전환 후 다양한 조인 연산을 수행하여 서브트리들을 추출하는 과정을 거친다. 본 연구는 기존 제시된 레이블 표현법[14, 15]에 따라 트리를 변환 후 의미를 고려한 유사성을 계산하여 마이닝을 수행하는 방안을 제시하고자 한다.

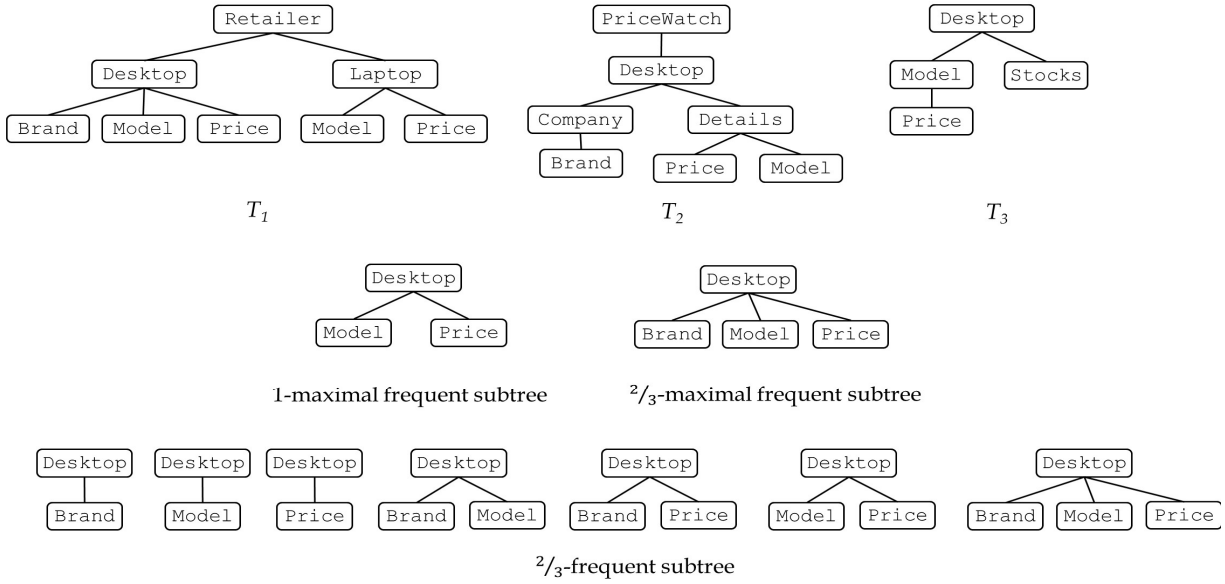

Fig. 2. Maximal Frequent Subtrees vs. Frequent Subtrees

1. Semantic Considered Bits representation

각각의 트리를 일련의 이진코들, 비트열(bit-string), 로 전환하는 표현 방식으로 기존 노드 레이블의 수에 따라 비트 수를 정한 방법[15]을 수정하여 레이블의 의미까지도 고려하여 비트화하는 방식을 본 논문에서는 제안하고자 한다.

트리 데이터베이스 D를 구성하는 전체 트리들의 노드 레이블 집합을 L이라 할 때, 각 레이블 $\ell \in L$ 은 k-비트 이진코드에 대응되며, 이때 이진코드 길이 n은 집합 L의 크기에 의해 정해지기 때문에 $\lceil \log_2 |L| \rceil$ 로 계산된다. 여기에 본 연구에서는 동음이의어 고려를 위한 2비트를 최상위 비트로 추가하여 (k+2)-비트로 정한다. 동일 단어에 대해 4개까지의 서로 다른 의미를 구분할 수 있는 시맨틱 사전이 선 구현된다는 가정 하에 트리의 레이블들은 (1) 시맨틱 사전 스캔을 통해 동음이의어 여부 판별 후 (2) 아니라면, 최상위 비트는 00으로 동음이의어라면 최상위 비트는 01, 10, 11 중 하나로 정한다. 서로 다른 노드 레이블들은 모두 동음이의어라는 의미를 반영하여 k+2 비트 길이의 이진코드가 할당된다.

루트부터 각 말단 노드에 해당하는 개별적인 경로들에 위치한 k-비트 코드들을 연접하여 경로당 하나의 비트열을 생성한다. 즉, 트리 구조의 최하위 레벨에 위치한 말단 노드들의 개수만큼 비트열이 생성된다.

(정의 7) 경로식

f개의 말단 노드들을 갖는 임의의 트리 T의 경로들의 집합 $P_T = \{p_1, p_2, \ldots, p_f\}$일 때, 임의의 경로 $p_e = \langle v_{e_1}, v_{e_2}, \ldots, v_{e_m} \rangle$는 루트노드부터 단일 말단노드까지 일련의 노드들의 연접리스트로 $1 \le e \le f$, $1 \le m \le |T|$이다. $|T|$는 트리의 사이즈로 전체 노드 수이다.

(정의 8) 비트열 경로식

트리 T의 각각의 노드 레이블에 대해 $k = \lceil \log_2 |L| \rceil + 2$ 인 k-비트 이진코드 매핑 후 개별 경로식은 깊이우선탐색 방식으로 인접 노드 간 이진코드 연접연산(·), $v_1 \cdot v_2 = k_{l_{v1}} \cdot k_{l_{v2}} = k_{l_{v1}} k_{l_{v2}}$, 이 이루어지며 비트열 경로식을 생성한다. 이진코드 연접연산에 의해 f개의 경로를 갖는 집합 P_T는 f개의 비트열 집합 $BP_T = \{bs_1, bs_2, \ldots, bs_f\}$로 변환된다.

트리 데이터베이스 D는 비트열 집합인 $BD = \{BP_{T_i} | 1 \le i \le n\}$로 변환 저장되며 일차적으로 수행되는 작업은 특정 기준 수치 이상 자주 발생하는 k-비트 이진코드들을 추출하는 것이다. 깊이우선탐색 방식에 의해 연접된 k-비트 코드들은 최상위 k-비트는 루트 노드이며 최하위 k-비트는 말단 노드에 해당한다. 모든 비트열들에 대해서 최상위부터 k-비트씩 스캐닝을 수행하며 트리들을 그룹화한다. Fig. 3은 Fig. 2에 보인 세 트리 중 하나인 T_3에 대해서 비트열 경로식으로 표현한 것이다. 시맨틱 코드에 해당하는 상위 2비트는 동음이의어들에 대한 그룹화를 수행하는 비트이기 때문에 앞으로의 계산 과정에서는 유도의 편의성을 위해 제외하고 표시한다.

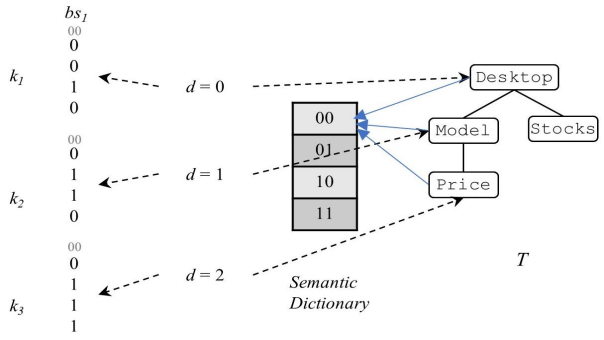

Fig. 3. Bits Representation Considered Labels Semantics

(정의 9) 페어 세트

트리 그룹화 구성에 사용되는 k-비트 코드들을 키라 하며 깊이에 따라 키 그룹 K_d, $0 \le d \le height(BD)$,를 생성한다. $height(BD)$는 가장 긴 비트열 bs_{height}를 k로 나눈 값이다. 키들은 깊이 d에서 자신들의 코드를 포함하고 있는 트리 인덱스 리스트를 값(tlist)으로 갖는 (key, tlist) 쌍을 구성하며 이를 k-비트 페어라 정의한다. 동일 키 그룹에 속하는 키들이 생성한 k-비트 페어를 깊이 d의 k-비트 페어 세트라 하고 $[P]_d$로 표기한다. 깊이별로 k-비트 페어 세트 생성이 완료되면 이후 마이닝의 모든 과정은 $[P]_d$를 통해서 이루어진다.

2. Frequent and candidate k-bit pairset

초기 k-비트 페어 세트 $[P]_d$의 모든 키들은 원본 D를 구성하는 트리들의 모든 레이블에 대응하는 것으로써 apriori 속성[ ]에 따라 특정 빈도수를 기준으로 키들을 분류한다. 즉, $[P]_d$에 속하는 모든 k-비트 키에 대해서 (정의 5)에 근거하여 $minsup$을 만족하는지 여부를 계산한다.

(정의 10) 빈발 k-비트 페어와 빈발페어세트

깊이 d에서의 임의 페어 (k, tlist) $\in [P]_d$ 의 |tlist| $\ge$ $minsup \times |D|$면 키 k는 빈발키이며 페어 (k, tlist)는 빈발하다고 한다. 빈발페어는 $[F]_d$로 분류되며 빈발하지 않은 페어는 $[C]_d$로 분류되는데 $[F]_d$는 빈발페어세트라 칭하며 $[C]_d$는 후보페어세트라 칭한다.

깊이별 초기 페어세트 $[P]_d$에 속하는 모든 (key, tlist) 페어들은 $minsup$에 의해 $[F]_d$와 $[C]_d$로 분류된다. $[F]_d$의 페어들의 모든 키들은 $minsup$을 만족하는 단일 노드 레이블에 해당하는 값들로 Apriori 방식을 따를 경우 빈발 단일 노드 레이블 간의 조인을 통해서 빈발 서브트리로 확장을 수행하게 된다. 그러나, 본 논문에서 제안하는 방식은 복잡한 조인연산 없이 후보페어세트 사용을 통해서 최대빈발 서브트리를 추출한다. Fig. 4는 Fig. 2의 T_1, T_2, T_3를 비트화 한 후 페어 세트들을 저장 후 페어 세트 $[P]_0$, $[P]_1$, $[P]_2$, $[P]_3$로부터 $minsup$ = 2/3를 적용한 빈발페어세트와 후보페어세트로 구분되어 저장된 결과를 보인다.

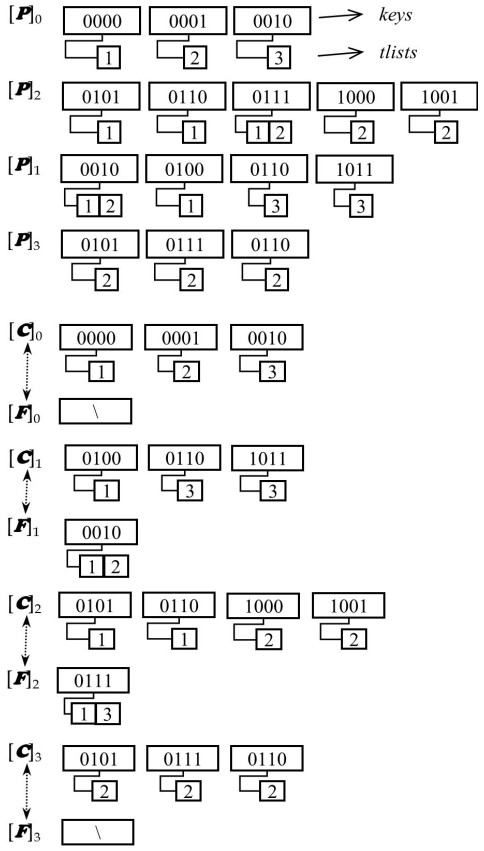

Fig. 4. Pairset, Candidate Pairset, and Frequent Pairset

깊이에 따라 분류된 빈발페어세트와 후보페어세트는 트리의 특성인 계층 구조와 (정의 3) 이 반영되지 않은 중간 결과이다. 즉, 빈발페어세트에 포함된 페어일지라도 다른 깊이에서는 후보페어세트로 분류될 수 있다는 점과 트리에서 여러 번 발생하나 다른 깊이에 위치했기에 때문에 서로 다른 후보페어세트에 위치해 있는 페어들은 합산하여 빈발페어로 간주해야 되는 부분이다. 이러한 연산을 위해 본 논문에서는 빈발페어세트와 후보페어세트의 모든 페어들 사이에 수행되는 cross-pairset 연산을 정의한다.

(정의 11) Cross-pairset 연산

빈발페어세트 집합 FP = {$[F]_0$, $[F]_1$, ..., $[F]_d$}와 후보페어세트집합 CP = {$[C]_0$, $[C]_1$, ..., $[C]_d$}가 주어졌을 때, cross-pairset 연산 1회(round)는 다음 두 단계로 구성된다:

제거 단계(pruning phase): ($[C]_{d-1}$ vs. $[F]_d$)와 ($[C]_d$ vs. $[F]_{d-1}$ 부터 $[F]_0$) 차이. d의 값은 $max(BD)$, $max(BD)$-1, ..., 1까지

감소하면서 깊이별 빈발 및 후보페어세트를 선정한다.[2)]

병합 단계(merging phase): ($[C]_h$ vs. $[C]_{h-1}$) 합집합. d의 값은 $max(BD)$, $max(BD)-1$, ..., 1까지 감소하면서 깊이별 후보페어세트를 선정한다.

제거 단계는 빈발페어세트에 포함되어 있는 페어들을 후보페어세트에서 제거하는 연산으로 이 때, 해당 페어의 tlist들은 빈발페어에 추가된다. 병합 단계는 apriori 기법에 대응하는 단계로 조인연산 없이 후보군을 생성한다. (정의 3)에 근거한 서브트리 특징을 반영하여 $[C]_h$와 $[C]_{h-1}$에 동일 키 페어가 존재하고 합한 tlist가 minsup을 만족한다면 이는 빈발 페어로 간주된다. 해당 페어는 $[F]_h$에 포함되고 남은 $[C]_h$와 $[C]_{h-1}$의 페어들은 $[C]_h$로 합해진다. 연산 cross-pairset을 통해 빈발페어세트들은 확장되며 최종적으로 완성된 빈발페어세트로부터 최대 빈발서브트리가 유도된다. Fig. 5는 cross-pairset 연산 수행 후 빈발 및 후보페어세트의 최종 결과이다.

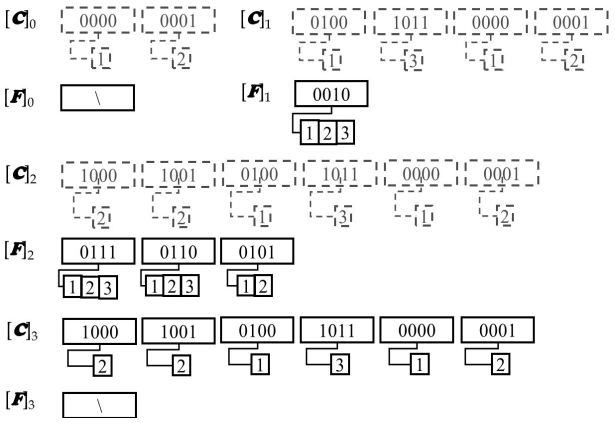

Fig. 5. Result of Cross-pairset Operation

3. Derivation of Maximal Frequent Subtree

최종 빈발페어세트로부터 최대 빈발서브트리 유도를 위해 고려해야 할 사항 두 가지가 존재한다. 첫째는, 최종 빈발페어세트에서 어떤 깊이의 빈발페어세트는 원소 즉, 페어가 존재하지 않을 수 있다. Fig. 4를 보면 $[F]_0$과 $[F]_3$이 해당 경우로 해당 깊이에서는 *minsup*을 만족하는 키 다시 말하면 레이블이 없었다는 것을 의미하기 때문에 최대 빈발서브트리 유도에 고려할 필요가 없다는 것이다. 둘째는, 트리가 계층 구조를 갖지만 이것이 빈발페어세트에 존재하는 모든 페어들이 서로 계층으로 연결되어 있다는 것을 의미하지 않는다는 것이다. 즉, 서로 다른 깊이의 빈발페어세트의 원소로 존재하는 페어 사이에 간선 연결 여부를 판별해야 한다는 것이다. 이는 매우 중요한 고려 사항으로 본 논문에서는 간선 여부 판별을 위해 *minsup*을 사용한다. Fig. 6의 결과를 보면 Fig. 5의 정제된 빈발페어세트로부터 깊이 1에서는 3개의 트리 인덱스를 갖는 0010이 그리고 깊이 2에서는 3개의 트리 인덱스를 포함하는 0111과 0110이, 2개의 인덱스를 갖는 0101이 존재한다. 트리 인덱스의 교집합 개수는 최소 2개로 (0010, 0111), (0010, 0110), 그리고 (0010, 0101) 모두 *minsup*을 충족하는 간선을 형성한다.

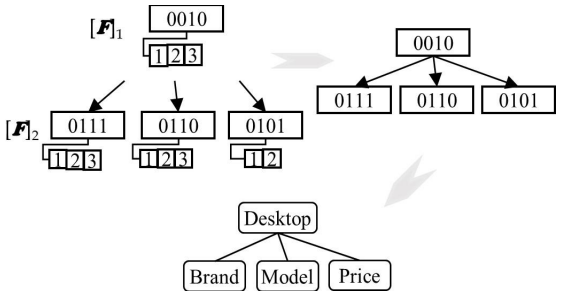

Fig. 6. 2/3-Maximal Frequent Subtrees from the Final Frequent Pairset

주어진 *minsup*을 충족하는 최대 빈발서브트리를 유도함에 있어서 다음 세 경우 중 하나의 의해 모든 빈발서브트리들이 생성된다는 것이 증명된다.

(정리) 트리 집합 D 가 주어지면 유도되는 최대 빈발서브는 모든 빈발서브트리들을 생성한다.

증명. 공백이 아닌 임의의 빈발페어세트 $[F]_i$와 $[F]_j$, $0 \le i < j \le d$, 에 대해 $[F]_i = \{(k^a, tlist_{k^a})\}$, $[F]_j = \{(k^b, tlist_{k^b}), (k^c, tlist_{k^c})\}$라 하자. $tlist_{k^b}$의 트리 인덱스는 x이고 $tlist_{k^c}$의 트리 인덱스는 $y(\neq x)$이며 $|tlist_{k^a} \cap tlist_{k^b}|$ 또는 $|tlist_{k^a} \cap tlist_{k^c}|$의 값이 $minsup \times |D|$ 보다 크다고 할 때 세 경우로 증명된다.

경우 1 ($x \in tlist_{k^a} \wedge y \notin tlist_{k^a}$) k^b는 k^a의 유일한 자식 노드이기 때문에 노드 집합 $N = \{k^a, k^b\}$, 간선 집합 $E = \{(k^a, k^b)\}$를 갖는 서브트리 S가 생성된다.

경우 2 ($y \in tlist_{k^a} \wedge x \notin tlist_{k^a}$) k^c는 k^a의 유일한 자식 노드이기 때문에 노드 집합 $N = \{k^a, k^c\}$, 간선 집합 $E = \{(k^a, k^c)\}$를 갖는 서브트리 S가 생성된다.

경우 3 ($x \in tlist_{k^a} \wedge y \in tlist_{k^a}$) k^b 와 k^c 모두 k^a의 자식 노드이기 때문에 노드 집합 $N = \{k^a, k^b, k^c\}$, 간선 집합 $E = \{(k^a, k^b), (k^a, k^c)\}$를 갖는 서브트리 S가 생성된다.

2) 임의의 깊이 h ($0 \le h \le d$)에 대해서 cross-pairset 연산 1회 수행 의미

Algorithm 1 *genBitSeq*

Input: T, L Output: a set of bits sequences BP

```
(1)  bs ← ⟨⟩                                  ▷initialize a variable for a bits sequence
(2)  for each path p ∈ T
(3)      bs ← makeSequence(r, bs)             ▷make a bits sequenced path from a root r
(4)      BP ← ⋃_{p∈T} bs
(5)  return BP

makeSequence(node u, current_bs cb)
(6)  k ← Semantic_Dictionary(u) · L(u)        ▷binary code is generated from a string label considered semantics
(7)  if u has no child node                   ▷current node is a leaf node
(8)      cb ← cb + k
(9)      return cb
(10) else                                     ▷current node is an internal node
(11)     cb ← cb + c
(12)     for each node v ∈ u' adjacent nodes
(13)         if v is not exploited
(14)             makeSequence(v, cb)
```

Fig. 7. Algorithm for Generating Bits Sequences Considered Semantics

4. Algorithms

본 연구에서 제안하는 방법의 주요 아이디어는 계층 구조를 갖는 트리의 모든 레이블과 경로를 동음이의어라는 시맨틱을 고려한 이진코드화하여 k-비트 코드와 트리 인덱스 하나의 쌍으로 하는 페어세트 구조에 저장하여 cross-pairset이라는 연산을 수행하여 임의의 최소지지도를 만족하는 모든 빈발트리에 대한 정보를 포함하고 있는 최대빈발서브트리를 유도하는 것이다. 이를 위한 주요 알고리즘들을 제시하고 그에 대한 분석을 수행한다. Fig. 7은 모든 트리들의 각 경로에 대해 레이블의 시맨틱을 반영하여 이진코드 비트열 경로로 변환하는 알고리즘을 보인다. 재귀함수인 *makeSequence*에 깊이우선탐색 방식에 따라 각 노드들이 비트화를 형성한다. 두 번째 알고리즘 *comFreSet*은 각 비트열 경로에 대해 깊이별로 (key, tlist) 쌍으로 구성된 페어세트 구조를 생성 후 *minsup*에 따라 일차적으로 빈발페어세트와 후보페어세트로 나눈다. 그 후 계층적 구조인 트리의 특성 반영을 위해 고안된 cross-pairset 연산을 수행하여 빈발페어세트를 확장하는 동시에 후보페어세트를 정제한다. 마지막 알고리즘 *maxSubtree*는 최종으로 도출된 빈발페어세트의 페어들에 대해서 트리 간선 연결을 위해 *minsup*을 적용한 재귀함수 *makeLink*를 통해 최대빈발서브트리를 유도한다. Fig. 8와 Fig. 9 은 순차적으로 두 알고리즘을 보인다.

본 연구에서 제안하는 알고리즘 적용을 위해 영화 예매를 위한 챗봇을 일차적으로 구현하였다. 그림 11은 해당 챗봇에 대한 화면이다. 연관단어 유추에 의해 사용자가 "램프의 요정 지니"라고 언급해도 정확히 알라딘 영화에 대한 예매를 완료하는 것을 확인할 수 있다. 그러나, 이는 단순 연관단어에 기반한 예매로 단어의 시맨틱 자체를 파악하고 예매를 수행하는 것은 아니다. 동음이의어, 동의이의어 등의 단어 유추를 위해서는 시맨틱 사전에 의한 알고리즘 적용이 선수행되어야 하는 것이며 이를 위해 본 논문에서 제시한 알고리즘 적용으로 문제 해결을 하고자 한다.

V. Conclusions

데이터중심 IoT 플랫폼의 궁극적인 목적은 생성된 빅데이터를 적시적소에 맞게 가공 및 분석을 수행하여 가치 있는 결과를 도출하여 최적의 솔루션을 제시하는 것이다. 이는 대량의 데이터를 분석하는 효율적이고 정확한 알고리즘에 기인한다. 본 논문은 분산되어 생성되는 대량의 비정형 데이터들을 트리라는 구조를 형성한다는 근거를 바탕으로 처리에 있어서 빠른 속도와 적은 공간을 차지하는 이진코드로 변형하여 저장 후 동음이의어라는 시맨틱을 고려하여 빈번하게 발생하는 데이터를 추출하는 방법을 제시하고 이에 대한 알고리즘을 수립하였다. 현재 해당 일고리즘의 실질적인 적용을 위해 영화 예매를 위한 챗봇을 일

Algorithm 2 *comFreSet*

Input: $\mathcal{BD}$, *minsup* Output: a set of all frequent pairsets $\boldsymbol{FP}$

(1) d $\leftarrow$ $height(\mathcal{BD})$ ▷*obtain the highest depth*
(2) **for** i $\leftarrow$ 0 to d ▷*generate pairset w.r.t. the depth*
(3) create PairSet $[\mathbf{P}]_i$
(4) classify $[\mathbf{P}]_i$ into $[\mathbf{F}]_i$ and $[\mathbf{C}]_i$ by the *minsup*
(5) delete $[\mathbf{P}]_i$ from the memory
(6) $FP \leftarrow FP \cup [\mathbf{F}]_i$
(7) **for** i $\leftarrow$ 0 to d ▷*do cross-pairset operations*
(8) find additional tree indexes by $([\mathbf{C}]_{i-1}, [\mathbf{F}]_i)$, and ▷*pruning step*
delete corresponding elements from $([\mathbf{C}]_{i-1})$
(9) **for** ℓ $\leftarrow$ $i-1$ to 0 ▷*pruning with previous frequent pairsets*
(10) find additional tree indexes by $([\mathbf{C}]_i, [\mathbf{F}]_\ell)$, and
delete corresponding elements from $([\mathbf{C}]_i)$
(11) find additional $(key, tlist_{key})$ pairs satisfying the *minsup* ▷*merging step*
by $(\mathbf{C}]_i, [\mathbf{C}]_{i-1})$, and delete corresponding elements from $[\mathbf{C}]_i$ and $[\mathbf{C}]_{i-1}$
(12) $[\mathbf{C}]_i \leftarrow [\mathbf{C}]_{i-1} \cup [\mathbf{C}]_i$
(13) return FS ▷*final FS has the results of cross-pairset*

Fig. 8. Algorithm for Computing Pairsets

Algorithm 3 *maxSubtree*

Input: FS, *minsup* Output: all maximal frequent subtrees MFS

(1) count $\leftarrow$ num(FS) ▷*number of frequent pairsets in FS*
(2) **for** ℓ $\leftarrow$ 0 to count ▷*search the first non-empty frequent PairSet*
(3) if $|K_\ell|$ of $[\mathbf{F}]_\ell$ is not zero ▷K_ℓ *is a set of keys in* $[\mathbf{F}]_\ell$
(4) first $\leftarrow$ ℓ, next $\leftarrow$ $\ell+1$
(5) break
(6) **for each** $key \in [\mathbf{F}]_{first}$ ▷*set roots of frequent subtrees*
(7) create a root whose label is *key*
(8) flag $\leftarrow$ *false*
(9) for ℓ $\leftarrow$ next to count
(10) **for each** $key \in [\mathbf{F}_\ell]$
(11) flag $\leftarrow$*makeLink*(key, ℓ) ▷*build explicit tree structure*
(12) if flag = false
(13) create a root whose label is *key*
(14) convert all binary codes to their original string labels

makeLink(key k, whichSet ws)

(15) pre $\leftarrow$ $ws-1$ ▷*set previous frequent pairset*
▷*check if there is any relation between k and other keys in ws*
(16) if $\exists\ tlist_{k' \in [\mathbf{F}]_{ws}} |tlist_{k'} \cap tlist_k| \geq minsup$
(17) create a node v whose label is k
(18) v becomes a child of the node whose label is k'
(19) return *true*
(20) else
(21) *makeLink*(k, pre)

Fig. 9. Algorithm for Deriving Maximal Subtrees

Fig. 10. Chat-bot for booking movie tickets

차적으로 구현하여 예매 관련 데이터를 생성하는 간단한 앱을 완성하였다. 이 앱의 문제는 시맨틱을 반영하지 못해 단순 키워드에 근거하여 예매를 수행한다는 것으로 본 연구는 이 문제 해결을 위해 연구에서 제안하는 알고리즘을 적용하여 구현하고 있다.

ACKNOWLEDGEMENT

This paper was supported by the Research Fund, 2018, Pyeongtaek University in Korea.

REFERENCES

[1] Jae-Ho Kim, Jae-Seok Yun, Seong-Chan Choe, and Min-U Ryu, "IoT Platform Technology Trends and Developments," Information and Communications Magazine, 30(8), 29-35, July 2013.

[2] Teik-Boon Tan and Wai_Khuen Cheng, "Software Testing Levels in Internet of Things (IoT) Architecture," 23rd International Computer Symposium, ICS 2018, Communications in Computer and Information Science 1013, pp. 385-390, Yunlin, Taiwan, December 2018. DOI: 10.1007/978-981-13-9190-3_40.

[3] Yu Liu, "A Data-centric Internet of Things Framework Based on Public Cloud," Linköping Studies in Science and Technology Licentiate Thesis No. 1850, September 2019. DOI: 10.3384/lic.diva-159770.

[4] J. Paik, "Weighted or Non-Weighted Negative Tree Pattern Discovery from Sensor-Rich Environments," Intelligent Automation And Soft Computing, Vol. 26, No. 1, pp. 193-204, March 2020. DOI: 10.31209/2019.100000140.

[5] Y. Chi, S. Nijssen, R. R. Muntz, and J. N. Kok, "Frequent Subtree Mining – an Overview," Fundamental Informaticae, Vol. 66, No. 1-2, pp. 161-198, November 2004.

[6] M. J. Zaki, "Efficiently Mining Frequent Trees in a Forest: Algorithms and Applications," IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 8, pp. 1021-1035, August 2005, DOI: 10.1109/TKDE.2005.125

[7] J. Paik, J. Nam, U. M. Kim, and D. Won, "Method for Extracting Valuable Common Structures from Heterogeneous Rooted and Labeled Tree Data," J. of Information Science and Engineering, Vol. 30, No. 3, pp. 787-817, March 2014.

[8] A. Inokuchi, T. Washio, and H. Motoda, "An Apriori-based Algorithm for Mining Frequent Substructures from Graph Data," Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery, LNCS, Vol. 1920, pp. 13-23, Lyon, France, September 2000, DOI: 10.1007/3-540-45372-5_2

[9] M. Kuramochi and G. Karypis, "Frequent Subgraph Discovery," *Proceedings of IEEE International Conference on Data Mining*, pp. 313-320, San Jose, California, 29 Nov.-2 December 2001, DOI: 10.1109/ICDM.2001.989534.

[10] Harmanjit Singh and Richa Sharma, "Role of Adjacency Matrix & Adjacency List in Graph Theory," International Journal of Computers & Technology , Vol. 3, No. 1, pp. 1021-1035, August 2012, DOI: 10.24297/ijct.v3i1c.2775

[11] Fengying Li, Enyi Yang, Anqiao Ma, Rongsheng Dong, "Optimal Representation of Large-Scale Graph Data Based on Grid Clustering and K2-Tree," Mathematical Problems in Engineering, Vol. 2020, Article ID 2354875, 8 pages, January 2020, doi.org/10.1155/2020/2354875.

[12] T. Miyahara, T. Suzuki, T. Shoudai, T. Uchida, K. Takahashi, and H. Ueda, "Discovery of Frequent Tag Tree Patterns in Semistructured Web Documents," *Proceedings of the 6th Pacific-Asia Conference of Advances in Knowledge Discovery and Data Mining*, LNCS, Vol. 2336, pp. 341-355, Taipei, Taiwan, May 2002, DOI: 10.1007/3-540-47887-6_35

[13] Kunihiko Sadakane and Gonzalo Navarro, "Fully-Functional Succinct Trees," Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, pp. 134-149, Austin, Texas, USA, January 2010, DOI: 10.1137/1.9781611973075.13

[14] Reuven Cohen, Pierre Fraigniaud, David Ilcinkas, Amos Korman, and David Peleg, "Labeling Schemes for Tree Representation," 7th *International Workshop on Distributed Computing IWDC 2005*, LNCS, Vol. 3741, pp. 13-24, Kharagpur, India, December 2005, DOI: doi.org/10.1007/11603771_2

[15] J. Paik, J. Nam, U. M. Kim, and D. Won, "Fast Extraction of Maximal Frequent Subtrees Using Bits Representation," J. of Information Science and Engineering, Vol. 25, No. 2, pp. 435-464, March 2009, DOI: 10.1.1.423.292.

Author

Juryon Paik received the B.E. degree in Information Engineering from Sungkyunkwan University, Korea in 1997. She worked at the Samsung SDS company about a year. She achieved her M.E. and Ph.D. degrees in Computer Engineering from the sane university in 2005 and 2008, respectively. Dr. Paik joined the faculty of the Department of Digital Information and Statistics at Pyeongtaek University, Pyeongtaek-si, Korea, in 2016. He is currently an assistant professor. She is interested in tree mining, big data mining, information retrieval and deep learning.